## **High Availability Web App with ALB, Route 53, and ASG**

### **Objective**

Deploy a **highly available, auto-scalable web application** behind an **ALB**, with a custom **Route 53 domain name**, and simulate dynamic scaling using CPU stress testing.

## **Assignment Requirements**

### **Part 1: Setup**

1. **Create a custom Route 53 Hosted Zone** (e.g., yourdomain.dev)
2. **Launch a Launch Template** with:  
   * Amazon Linux 2
   * httpd + stress preinstalled via User Data
   * index.html shows hostname and instance ID
3. **Create a Target Group** with HTTP health checks on /

### **Part 2: Auto Scaling Group (ASG)**

1. Create an **ASG** using:  
   * The launch template
   * Spread across **2 AZs**
   * **Min = 1**, **Max = 4**, **Desired = 2**
   * Attach to the above **Target Group**
2. Add a **Target Tracking Scaling Policy**:  
   * Scale out if **average CPU > 50%**
   * Set **warm-up** appropriately

**Purpose**: Tells Auto Scaling to wait before including a newly launched instance in metrics.

**Parameter**: EstimatedInstanceWarmup (e.g., 300 seconds)

1. Add tags:  
   * owner = your-name
   * Name = yourname-alb-asg-instance

### **Part 3: ALB Configuration**

1. Create an **Application Load Balancer**:  
   * Internet-facing, IPv4
   * Listener on **HTTP 80**
   * Forward to your ASG target group
2. **Optional**: Add HTTPS listener and use ACM certificate

### **Part 4: Route 53**

1. Create an **Alias A Record** in Route 53:  
   * E.g., app.yourdomain.dev → ALB DNS name
2. Test domain resolution and ALB routing

### **Part 5: Stress Test & Scaling Observation**

Use stress in user-data or manually SSH to:  
  
stress --cpu 1 --timeout 300

1. Monitor:  
   * **CloudWatch metrics**
   * **ASG Activity**
   * **ALB target health**
2. Wait for scale-out to **Max = 4 (couldn’t get it to scale past 3…)**
3. Then allow it to scale back in after stress ends

## **Deliverables**

* Architecture diagram (draw.io / Lucidchart / hand-drawn)
* Shell script to automate launch template and ASG creation
* Screenshot of:  
  1. Route 53 record
  2. ALB DNS and health check
  3. ASG scaling activity
  4. top output during stress
* Answers to:  
  1. What termination policy did you use and why?  
     The one that spins up new instances while the other ones are still terminating. Used this to minimize the wait time for new instances to be usable.
  2. How does warm-up time affect your scaling behavior?  
     It will wait until the time is up to create new instances.
  3. How would you migrate this to a Spot/On-Demand mix  
     Terminate all instances, go back to the ASG and update the policy to be mixed. You can also change this on the launch template.
* Google doc with github repo link